IN THE CLAIMS

Claims 1-34 were pending at the time of the Office Action.

Claims 1-34 are currently pending.

Claims 35-42 were previously withdrawn.

Claims 1-4, 10-13, 17, 21-22, 25-29, 31-34 are currently amended.

1. (Currently Amended) A method of selecting at least one digital media component to construct a device that accomplishes one or more tasks identified in an extensible profile, comprising:

retrieving, from the extensible profile, at least one required capability for performing the selected task as requested by an application;

selecting, from a component register, one or more component entries with capability lists that include the required capability; and

instantiating one or more components corresponding to the selected entries, wherein instantiating occurs via an Application Programming Interface (API),

wherein the application uses the extensible profile to determine which of the one or more components are needed for the selected task, and

wherein the extensible profile may be updated or modified without modifying the application.

2. (Currently Amended) The method of claim 1, wherein retrieving from the extensible profile at least one required capability for performing the selected task comprises:

receiving a request to perform a selected task; and

searching an extensible profile for one or more entries corresponding to the selected task.

3. (Currently Amended) The method of claim 1, wherein [[the]] a profile register comprises at least one ~~a plurality of~~ extensible profile ~~data structures~~, and wherein each extensible profile ~~data structure~~ comprises a key that identifies a task.

4. (Currently Amended) The method of claim 3, wherein [[the]] an ~~plurality of~~ extensible profile ~~data structures~~ comprises at least one subprofile entry, wherein the subprofile entry identifies a capability required to perform the task associated with the profile entry.

5. (Original) The method of claim 4, wherein the subprofile entry comprises:

a subprofile identifier that uniquely identifies the subprofile entry; and

one or more operating parameters associated with the function.

6. (Original) The method of claim 4, wherein selecting one or more entries from a component register that includes a capability list with the required capability comprises searching a component register for entries with capability lists comprising an identifier equal to one or more subprofile identifiers associated with the selected task.

7. (Original) The method of claim 6, wherein selecting, from a component register, one or more entries whose capability lists include the required capability comprises searching a component register entry's capability list for entries comprising:

an identifier equal to one or more subprofile identifiers associated with the selected task; and

operating parameters compatible with the operating parameters specified in the subprofile.

8. (Original) A computer readable media having computer-readable instructions thereon which, when executed by a computer, implement the method of claim 1.

9. (Original) A computer readable media having computer-readable instructions thereon which, when executed by a computer, implement the method of claim 6.

10. (Currently Amended) An apparatus, comprising:

a processor;

a memory module connected to the processor and comprising logic instructions operative to configure the processor to:

retrieve, from an extensible profile, at least one required capability for performing a selected task as requested by an application;

select, from a component register, one or more entries that include the required capability in their capability list; and instantiate via an application programming interface (API) one or more components corresponding to the selected

entries;

wherein the application uses the extensible profile to determine which of the one or more components are needed for the selected task,

wherein the extensible profile may be updated or modified without modifying the application.

11. (Currently Amended) The apparatus of claim 10, further comprising:

an interface for receiving a request to perform the selected task; and

logic instructions, executable on the process in response to receiving the request, for searching [[the]] a profile register for one or more entries corresponding to the selected task.

12. (Currently Amended) The apparatus of claim 10, wherein the profile register comprises at least one ~~a plurality of~~ extensible profile ~~data structures~~, and wherein each extensible profile ~~data structure~~ comprises a key that identifies a task.

13. (Currently Amended) The apparatus of claim 12, wherein the extensible profile ~~plurality of profile data structures~~ comprises at least one subprofile entry, wherein the subprofile entry identifies a capability required to perform the task associated with the profile entry.

14. (Original) The apparatus of claim 13, wherein the subprofile entry comprises:

a subprofile identifier that uniquely identifies the subprofile entry; and

one or more operating parameters associated with the function.

15. (Original) The apparatus of claim 13, wherein the logic instructions operative to configure the processor to select, from a component register, one or more entries whose capability lists include the required capability further comprise logic instructions operative to configure the process to search a component register for entries with capability lists comprising an identifier equal to one or more subprofile identifiers associated with the selected task.

16. (Original) The apparatus of claim 15, wherein the logic instructions operative to configure the processor to select, from a component register, one or more entries whose capability lists include the required capability further comprise logic instructions that configure the processor to search a component register for entries whose capability lists comprise:

an identifier equal to one or more subprofile identifiers associated with the selected task; and

operating parameters compatible with the operating parameters specified in the subprofile.

17. (Currently Amended) A method of interfacing digital media components on a computer-based processing device, comprising:

constructing a component register comprising of entries which contain listings of capabilities of digital media components accessible to the computer-based processing device; and

in response to a request from an application for digital media services, searching the component register for a component capable of providing the requested service,

wherein the application uses an extensible profile to determine which of the digital media components are needed for the selected task,

wherein the extensible profile may be updated or modified without modifying the application.

18. (Original) The method of claim 17, wherein constructing a component register with entries with lists of capabilities of digital media components accessible to the computer-based processing device comprises registering a digital media component.

19. (Original) The method of claim 17, wherein the component register comprises an entry for a plurality of digital media components registered with the computer-based processing device, wherein each entry comprises:

a first data field that identifies the digital media component;

one or more groups of fields, where

a data field that identifies a function performed by the digital media component; and

another data field that identifies one or more operational parameters with an associated function identified in the first data field in the group.

20. (Original) The method of claim 19, wherein the data fields are logically linked or stored in a common data structure.

21. (Currently Amended) The method of claim 17, further comprising constructing a profile register comprising at least one ~~record~~ extensible profile representing a digital media function.

22. (Currently Amended) The method of claim 21, wherein searching the component register for a component capable of providing the requested service comprises:

mapping the requested service onto the profile register to select an extensible profile corresponding to the service; and

mapping the selected extensible profile onto the component register to select one or more digital media components capable of providing the requested service.

23. (Original) The method of claim 17, further comprising instantiating the selected one or more components.

24. (Original) The method of claim 23, further comprising connecting the one or more instantiated components to other digital media components to form a device that performs a series of digital media tasks.

25. (Currently Amended) A method of interfacing digital media components on a computer-based processing device, comprising:

constructing a component register comprising at least one entry including listings of capabilities of digital media components accessible to the computer-based processing device, wherein at least one listing comprises one or more data fields, including:

a first data field that identifies a function performed by a digital media component; and

a second data field that identifies one or more operational parameters associated with a function identified in the first data field ;

constructing a profile register comprising at least one record representing a digital media function, the record comprising a data field having one or more operating parameters associated with the digital media function; and

in response to a request from an application for digital media services:

searching the profile register for a record that corresponds to the requested media service; and

searching the component register for a component capable of providing the requested service,

wherein the profile register is used to determine which of the digital media components are needed for the request from the application for digital media services,

wherein the profile register may be updated or modified without modifying the application for digital media services.

26. (Currently Amended) A method of assembling a topology of digital media components on a computer-based processing device, comprising:

reading lists of capabilities from a profile register;

searching a component register for entries containing the capabilities indicated in the profile register; and

rejecting components that lack the capabilities indicated in the profile register, or that have capabilities incompatible with the capabilities in the profile register,

wherein the profile register determines which components are needed for a selected task requested by an application,

wherein the profile register may be updated or modified without modifying the application.

27. (Currently Amended) The method of claim 26 further comprising:

instantiating one or more components; and

attempting to apply an extensible profile configuration to the instantiated component.

28. (Currently Amended) The method of claim 27, further comprising:

searching for additional components in the component register if the attempt to apply a extensible profile configuration to the instantiated component fails; and rejecting components that have capabilities incompatible with the capabilities in the profile register.

29. (Currently Amended) The method of claim 26, further comprising merging the extensible profiles's capability list with additional capabilities from a user or an application used in the search process.

30. (Original) The method of claim 29, wherein the additional capabilities include a vendor identification or certification identification.

31. (Currently Amended) A method of assembling and configuring a topology of digital media components on a computer-based processing device, comprising:

using a profile structure and one or more associated capability lists to select a component;

instantiating the selected component;

applying an extensible profile to the selected component; and

logically connecting the component to one or more additional components, wherein the extensible profile may be updated or modified without modifying a requesting application.

32. (Currently Amended) The method of claim 31, wherein the profile structure comprises a field that includes a list of mandatory settings; and wherein applying an

extensible profile to the selected component comprises generating a signal if the selected component cannot implement a mandatory setting.

33. (Currently Amended) The method of claim 32, wherein an application uses the signal to determine whether the extensible profile structure was implemented successfully.

34. (Currently Amended) A method of configuring a topology of encoding and multiplexing digital media components on a computer-based processing device, comprising:

searching an extensible profile for a multiplexer subprofile configuration;

searching a component register for a multiplexer object compatible with the multiplexer subprofile;

instantiating a multiplexer;

configuring the multiplexer by applying the subprofile configuration settings using an interface API;

connecting the multiplexer to an output of a content source, and, for each input stream of the multiplexer:

searching the extensible profile for an encoder subprofile;

searching the component register for a multiplexer object compatible with the subprofile;

configuring the encoder by applying the subprofile configuration settings using an interface API; and connecting the encoder to the multiplexer,

wherein the extensible profile determines which multiplexer objects are compatible with the multiplexer subprofile information,

wherein the extensible profile may be updated or modified without modifying a requesting application.

35. (Withdrawn) An API that implements a plurality of methods for controlling one or more devices via a plurality of control identifiers, wherein:

the control identifiers correspond to one or more configuration settings that have a defined dependency ordering that can be expressed as a directed acyclic dependency graph;

the configuration settings are structured such that changing a parameter causes a component to reconfigure one or more dependent settings; and

a high-level configuration settings can be modified independent of a low-level configuration setting.

36. (Withdrawn) The API of claim 35, wherein setting a high-level parameter causes one or more low-level parameters to be reconfigured, such that the low-level parameters are consistent with the high-level parameter.

37. (Withdrawn) The API of claim 35, wherein, in response to a change in a setting, a return parameter identifying the change in setting is made available on the control API.

38. (Withdrawn) The API of claim 37, comprising a method for enabling an application to register to receive an event from a component, wherein registering comprises providing a reference to the component.

39. (Withdrawn) The API of claim 35, wherein a change in a setting triggers an event notification that indicates the change in the setting.

40. (Withdrawn) An API that implements a plurality of methods for controlling one or more components via a plurality of control identifiers, comprising:

an interface method that enables a user to instruct a component to enable event detection and generation based on a GUID identifier; and

an interface method that enables the user to associate a custom data set to be returned with an event notification.

41. (Withdrawn) The API of claim 40, wherein the custom data set identifies a particular component, such that the data set can be used to correlate the event notification with a specific instance of the component.

42. (Withdrawn) A method of implementing one or more methods associated with anICodecAPI, wherein one or more configuration settings are defined using using KsPropertySets, and wherein remaining methods are defined from a group of settings defined by operations selected from the group of operations consisting of:

CODECAPI_SET ALL DEFAULTS, CODECAPI_ALLSETTINGS, CODECAPI_SUPPORTEVENTS, AND CODECAPI_CURRENTCHANGELIST.